CHAPTER 7

DIGITAL ELECTRONICS

BA3Y5

LEARNING OBJECTIVES

After the completion of this Chapter, the student will

1. **Understand the basic concepts of analog and digital signals**
2. **Understand the functions of digital circuit**
3. **Know about the number system**
4. **Understand the conversion of number system and basic of arithmetic operations**
5. **Develop the skill of converting one code format into other code formats**

INTRODUCTION

The branch of electronics, which deals with digital circuits, is called digital electronics. Over the past several decades, digital electronics have been utilized in the design and manufacturing of various industrial, commercial and household electronic gadgets. Due to the proliferation of digital electronics, it is very important to inculcate the basic knowledge of digital electronics to develop conceptual knowledge and practical experience among the stakeholders.

Electronic systems can be classified into two types of systems in which the mode of electron transfer from one end to another end differs. They are,

1. Analog system
2. Digital system

In electronics there are two types of signal.

1. Analog signals
2. Digital signals

7.1. ANALOG AND DIGITAL SIGNALS

i) Analog Signals

A continuously varying signal(voltage or current) is called as an analog signal.

Example: Sinusoidal waves.

A sample of analog signal that varies with time is shown in Figure 7.1.

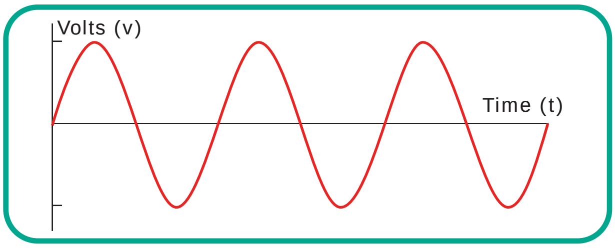


Figure7.1 Representation of an Analog signal

ii) Digital Signal

A signal (voltage or current) that can have only two discrete values is called a digital signal. Example: Square wave. The digital waveform is shown in Figure 7.2.

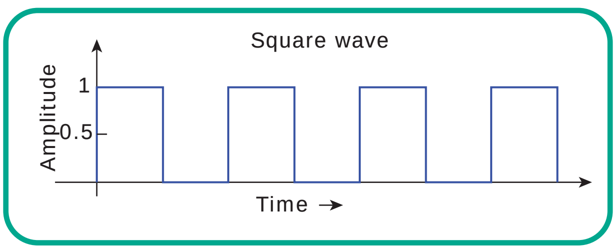


Figure 7.2 Representation of Digital Signal

Digital operations have two states (i.e. ON or OFF) and hence it is more simple and reliable than many valued analog operations.

7.2 DIGITAL CIRCUITS

An electronic circuit that handles only a digital signal is called a digital circuit.

Example: Digital calculator, Digital computer

The digital operation is a two state operation (i.e. ON or OFF, 1 or 0) and therefore a digital circuit uses only two digits 1 and 0 in the binary number system. In order to understand the concepts in digital circuits, First we discuss about the data representation and the number system in the following.

7.3 DATA REPRESENTATION

Computer handles data in the form of '0'(Zero) and '1' (One). Any kind of data like number, alphabet, special character should be converted to '0' or '1' which can be understood by the Computer. '0' and '1' that the Computer can understand is called Machine language. '0' or '1' are called 'Binary Digits'(BIT). Therefore, the study of data representation in the computer is important.

1. A bit is the short form of Binary digit which can be '0' or '1'. It is the basic unit of data in computers.
2. A nibble is a collection of 4 bits (Binary digits).
3. A collection of 8 bits is called Byte. A byte is considered as the basic unit of measuring the memory size in the computer.
4. Word length refers to the number of bits processed by a Computer's CPU. For example, a word length can have 8 bits, 16 bits, 32 bits and 64 bits (Present day Computers use 32 bits or 64 bits)

Computer memory (Main Memory and Secondary Storage)is normally represented in terms of KiloByte (KB) or MegaByte (MB). In decimal system, 1 Kilo

Table 7.1 Memory Size (Read 2^10 as 2 power 10)		
Name	Abbr	Size
Kilo	K	2^10=1,024
Mega	M	2^20=1,048,576
Giga	G	2^30=1,073,741,824
Tera	T	2^40=1,099,511,627,776
Peta	P	2^50=1,125,899,906,842,624
Exa	E	2^60=1,152,921,504,606,846,976
Zetta	Z	2^70=1,180,591,620,717,411,303,424
Yotta	Y	2^80=1,208,925,819,614,629,174,706,176

represents 1000, that is , 10^3 . In binary system, 1 KiloByte represents 1024 bytes that is 2^{10}. The following table represents the various memory sizes:

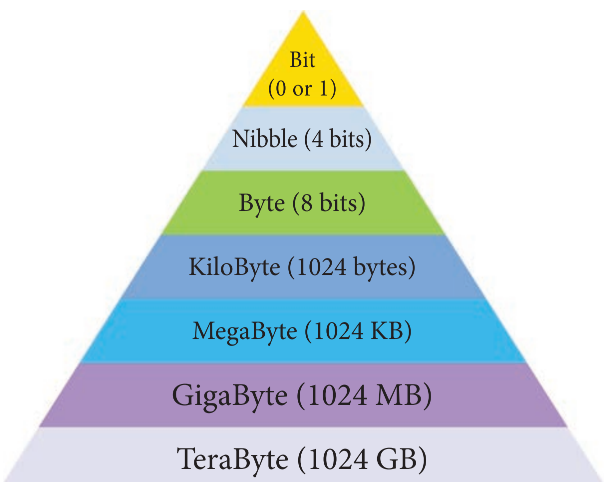


Figure 7.3 Data Representation

Bytes are used to represent characters in a text. Different types of coding schemes are used to represent the character set and numbers. The most commonly used coding scheme is the American Standard Code for Information Interchange (ASCII).

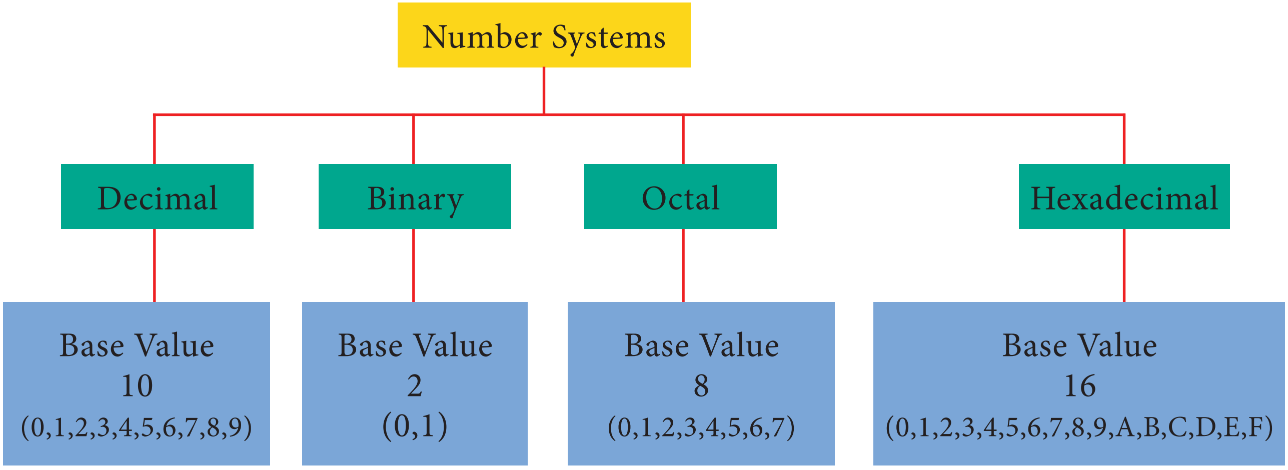
7.4 NUMBER SYSTEM

In digital electronics the number system is used for representing the information. It is commonly used to count any activity or articles. In practical life, we are using decimal number system.Other number systems are Binary, Octal and Hexadecimal number system. Each number system is uniquely identified by its base value or radix. Radix or base is the count of number of digits in each number system.

Computers, microprocessor and digital electronic devices do not process decimal numbers. Instead, they work with binary number, which use only the two digits'0' and'1'.

People do not like working with binary numbers, owing to their very lengthy combinations of digits, while representing larger decimal values.

As a result, octal and hexadecimal numbers are widely used to compress long strings of binary numbers. Types of number systems are given below.



7.4.1 Decimal Number System

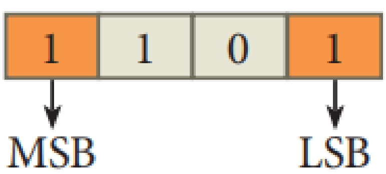
Decimal number consists of 0,1,2,3,4,5,6,7,8,9 (10 numbers). It is the oldest and most popular number system used in our day to day life. It has radix or base of 10.

Example : 345_{10}

7.4.2 Binary Number

Binary number contains only two numbers of '0' and '1'. It has radix or base of 2.

Example: 1101_2



The left most bit in the binary number is called as Most Significant Bit (MSB) and it has the largest positional weight.

The right most bit in the binary number is called as Least Significant Bit(LSB) and it has the smallest positional weight.

7.4.3 Octal Number System

Octal number system contains only eight numbers of 0,1,2,3,4,5,6 and 7. It has a radix or base of 7.

Example: 7612_8

7.4.4 Hexadecimal Number System

Hexadecimal number or Hex number system contains only sixteen numbers of 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E, and F. The first 10 symbols are the same as in the decimal system, 0 to 9 and the remaining 6 symbols are taken from the first 6 letter of the alphabet sequence, A to F, where A represents 10, B is 11, C is 12, D is 13, E is 14 and F is 15. It has a radix or base of 16.

Example: $508D_{16}$

The table shows the Binary, Octal,Hexadecimal equivalent of Decimal Numbers.

Decimal	Binary	Octal	Hexadecimal
0	0000	000	0000
1	0001	001	0001
2	0010	002	0002
3	0011	003	0003
4	0100	004	0004
5	0101	005	0005
6	0110	006	0006
7	0111	007	0007
8	1000	010	0008
9	1001	011	0009
10	1010	012	A

11	1011	013	B
12	1100	014	C
13	1101	015	D
14	1110	016	E
15	1111	017	F

7.4.5 BCD (Binary Coded Decimal) Number

A nibble is a string of 4 bits. BCD numbers express each decimal digit as nibble. It is a decimal number represented in binary form with 0 and 1. The lowest number is 0000 (0) and the highest number is 1001(9)

Example: $1000\ 0111_{BCD}$

Place Value

The binary, octal, decimal, and hexadecimal numbers are weighted numbers. Hence, every number system can be converted into any other number system through a process called conversion. After conversion, the weight of the number should not be varied. The weight of each number is represented as follows.

Decimal Number System

Number	2	8	5	7	.4	5
Weight of each digit	10^3	10^2	10^1	10^0	$.10^{-1}$	10^{-2}

Binary Number System

Number	1	0	1	1	.0	1
Weight of each digit	2^3	2^2	2^1	2^0	$.2^{-1}$	2^{-2}

Octal Number System

Number	7	3	5	6.	3	2
Weight of each digit	8^3	8^2	8^1	8^0	$.8^{-1}$	8^{-2}

Hexadecimal Number System

Number	8	A	B	5	. C	9
Weight of each digit	16^3	16^2	16^1	16^0	. 16^{-1}	16^{-2}

7.5 CONVERSIONS

Conversion of binary number from one number format to another number format can be performed by adapting some rules and regulations. Some of the important conversion processes are explained below. For the conversion of integer and fractional number, separate conversion methods are used.

7.5.1 Decimal to Binary Conversion

To convert Decimal to Binary "Repeated division by 2 method" can be used the decimal number is divided by 2, and writing down the remainder after each division. The remainders are taken in reverse order to form the binary number.

Example: Conversion of 26_{10} to its equivalent binary number

$$\begin{array}{r|l} 2 & 26 \\ 2 & 13-0 \\ 2 & 6-1 \\ 2 & 3-0 \\ & 1-1 \end{array}$$

Hence, $11010_2 = 26_{10}$

7.5.2 Binary to Decimal Conversion

To convert binary number to its equivalent decimal number, multiply each binary digit by its weight and then add the resulting products.

Example: Conversion of 1101_2 to its equivalent decimal number.

$$\begin{array}{cccc} 1 & 0 & 1 & 1 \\ 2^3 & 2^2 & 2^1 & 2^0 \end{array}$$

Equivalent decimal number

$= (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$

$= (1 \times 8) + (0 \times 4) + (1 \times 2) + (1 \times 1)$

$= 8 + 0 + 2 + 1 = 11$

Hence, $1011_2 = 11_{10}$

7.5.3 Decimal to Octal Conversion

To convert decimal to octal, "Repeated division by 8 method" can be used. decimal to octal conversion, the decimal number is divided by 8, and writes down the remainder after each division. The remainders are taken in reverse order to form the octal number.

Example: Conversion of the decimal number 408 to its equivalent octal number.

$$\begin{array}{r|l}
8 & 408 \\
8 & 51 - 0 \\
8 & 6 - 3
\end{array}$$

Hence, $408_{10} = 630_8$

7.5.4 Octal to Decimal Conversion

To convert an octal number to its equivalent decimal number, multiply each octal digit by its weight and then add the resulting products.

Example: Conversion of an octal number 375 into its equivalent decimal number.

The weight of 5 is 8^0, 7 is 8^1 and 3 is 8^2.

Hence, the equivalent decimal number is

$= (3x8^2)+(7x8^1)+(5x8^0)$

$= (3x64)+(7x8)+(5x1)$

$= 192+56+5= 253$

Hence, $375_8 = 253_{10}$

7.5.5 Decimal to Hexadecimal Conversion

To convert Decimal to Hexadecimal, "Repeated division by 16 method" can be used. The decimal number is divided by 16 and write down the remainder after each division. The remainders are taken in reverse order to form the hexadecimal number.

Example: Conversion of a decimal number 4538 to its equivalent hexadecimal number.

$$\begin{array}{r|l}
16 & 4538 \\
16 & 283 - 10 \\
16 & 17 - 11 \\
 & 1 - 1
\end{array}$$

Hence, $4538_{10} = 11BA_{16}$

7.5.6 Hexadecimal to Decimal Conversion

To convert the hexadecimal to its equivalent decimal number, multiply each hexadecimal digit by its weight and then add the resulting products.

Example: Conversion of a hexadecimal number of B35 to its equivalent decimal number.

The weight of B is 16^2, 3 is 16^1 and 5 is 16^0

Hence its equivalent Decimal number is

$= (Bx16^2)+(3x16^1)+(5x16^0)$

$= (11x256)+(3x16)+(5x1)$

$= 2816+48+5$

$=2869$

Hence, $B35_{16} = 2869_{10}$

7.5.7 Octal to Binary Conversion

In this conversion, each octal digit is converted into its equivalent three digit binary form. The octal number and its equivalent three digit binary numbers are shown in the Table 7.2.

Table 7.2: Conversion of Octal into Equivalent Binary Number	
Octal number	**Equivalent Binary number**
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Example: Conversion of an octal number 43 to its equivalent binary number.

4 3

100 011

$43_8 = 100011_2$

7.5.8 Binary to Octal Conversion

The binary numbers are grouped as 3-bit from left to right. If there is any binary digit left with one or two bits then sufficient numbers of zero are added to the left most side of the binary number. Then, grouped 3-bit number is converted into an equivalent octal number.

Example: Conversion of a binary number of 010111011 to its equivalent octal number.

$\underline{010}\ \underline{111}\ \underline{011}$

2 7 3

Hence, $010111011_2 = 273_8$

7.5.9 Hexadecimal to Binary Conversion

In this conversion, each hexadecimal digit is converted into its equivalent four digit binary form.

The hexadecimal number and its equivalent 4 digit binary numbers are shown in the Table 7.3.

Table 7.3: Conversion of Hexadecimal into Equivalent Binary Number	
Hexadecimal Number	**Equivalent Binary Number**
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Example: Conversion of a hexadecimal number 7B3 into its equivalent binary number.

7 *B* 3

0111 1011 0011

Hence, $7B3_{16} = 011110110011_2$

Note: Delete the left most zeros.

7.5.10 Binary to Hexadecimal Conversion

In this conversion, the binary number is arranged in group of 4 bits. Suppose the binary number grouping is not completed with the 4 digits, sufficient numbers of zero are added to the left most side of the binary number.

Example: Conversion of a binary number 110110101011100 into its equivalent hexadecimal number.

0110 1101 0101 1100

6 D 5 C

Hence, $110110101011100_2 = 6D5C_{16}$

7.5.11 Fractional Decimal to Binary Conversion

The method of "Repeated multiplication by 2" has to used to convert the decimal fractions.

0.2 × 2 = 0.4	0 (Step-1)
0.4 × 2 = 0.8	0 (Step-2)
0.8 × 2 = 1.6	1 (Step-3)
0.6 × 2 = 1.2	1 (Step-4)
0.2 × 2 = 0.4	0 (last integer part obtained)

Note: Fraction repeats, the product is the same as in the first step.

Write the integer parts from top to bottom to obtain the equivalent fractional binary number. Hence $(0.2)_{10} = (0.00110011\ldots)_2$

7.5.12 Decimal to BCD Conversion

In this Conversion, each decimal digit is converted into its equivalent 4 digits binary form (BCD).

Example: Conversion of a decimal number 892 to its equivalent BCD number.

8 9 2

1000 1001 0010

Hence, $892_{10} = 100010010010_{BCD}$

7.5.13 BCD to Decimal Conversion

In this method, each BCD number grouped in the form of 4 digit binary pattern is converted into its equivalent decimal number.

Example: Convert a BCD number100100111000 to its equivalent decimal number.

1001 0011 1000

9 3 8

Hence, $100100111000_{BCD} = 938_{10}$

7.6 BINARY ADDITION AND SUBTRACTION

A logic circuit can be used to perform arithmetic functions like addition, subtraction, multiplication, division etc. For performing these operations, compliment method of number patterns are used. First, we will see the complement methods in order to understand the basic arithmetic operations.

7.6.1 One's Complement Method

In one's complement method, each binary bit of the number is changed from 0 to 1 or 1 to 0 depending on the existing bit value.

For instance, the binary number is $A_3\, A_2\, A_1\, A_0 = 0010$, its corresponding one's complement number is $\bar{A}_3\, \bar{A}_2\, \bar{A}_1\, \bar{A}_0 = 1101$

The same principle will apply for number having any bit length and its corresponding one's complement number can be obtained by complement each bit.

7.6.2 Two's Complement Method

The two's complement of a binary number is the number that results when we add '1' to the one's complement number. The formula for two's complement of a binary number is given below.

Two's complement number = one's complement +1

For instance, to find the two's complement number of 0101, the following procedure is employed.

$0101 \Rightarrow 1010$ (1's *complement*)

$1010+1 \Rightarrow 1011$ (2's *complement*)

7.6.3 Binary Addition

For binary addition, the arithmetic rules used are given below.

1. 0+0=0
2. 0+1=1, No carry is formed
3. 1+0=1, i.e. carry=0
4. 1+1=0,with a carry of 1, and sum = 0 i.e. 1+1 =10_2.

This is a binary number10 and not the decimal number ten. Here, the first digit 0 is called sum and next digit 1 is called carry.

Examples:

1. Add the binary numbers 1011 and 1100

$$\begin{array}{r} 1011+ \\ 1100 \\ \hline 10111 \\ \hline \end{array}$$

Sum=0111, and Carry=1

2. Add the binary number 11101 with 11011001.

$$\begin{array}{lll} \text{The first number} & = & 00011101\ + \\ \text{The second number} & = & \underline{11011001} \\ \text{Result} & = & \underline{11110110} \end{array}$$

7.6.4 Binary Subtraction

The general rules for carrying out the binary subtraction are given below.

1. 0-0=0
2. 1-0=1
3. 1-1=0
4. 0-1=1,with a borrow of 1 from the next higher bit.
5. 10-1=1

Example: Subtract 0111 from 1011

$$\begin{array}{lll} 1011 & => & 11_{10} \\ 0111 & => & 7_{10} \\ \hline 0100 & => & 4_{10} \\ \hline \end{array}$$

First column => 1-1 =0

Second column =>1-1 = 0

Third column =>0-1 = 10-1 = 1

Forth column => After borrowing, the fourth column becomes 0

Hence, 0-0=0

$1011_2 - 0111_2 = 0100_2$

7.7 BINARY CODES

All digital circuits operate with only two states namely, High and Low or ON and OFF or 1 and 0. In binary number system, the number of bits required goes on increasing as the numbers become larger and larger. So, some special binary codes are required to represent alphabets and special characters. Based on these points, different types of binary code have been developed.

They are,

1. BCD codes
2. Gray codes
3. Excess 3 code
4. ASCII code

7.7.1 BCD - 8421 Code Conversion

A group of bits (usually four) which are used to represent decimal numbers 0 to 9 are called BCD(Binary Coded Decimal) codes. The most popular BCD code is 8421 code. The 8421 indicates the binary weights of the four bits (2^3, 2^2, 2^1, 2^0). Using the four bits with weights 8,4,2,1, we can easily represent the decimal numbers 0 to 9 as given in the Table 7.4.

Table 7.4: Conversion of Decimal Number into BCD Code

Decimal Numbers	BCD Code
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	0001 0000
56	0101 0110
963	1001 0110 0011

7.7.2 Gray Code

The gray code is not a weighted code. Therefore it is not suitable for arithmetic operations, but finds applications in input/output devices and in some types of analog to digital converters.

Table 7.5: Gray code conversion

Decimal numbers	Binary code	Gray code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

The gray code is a minimum change code in which only one bit in the code group changes when moving from one step to the next. The gray code is also called as reflected binary code, which has a special property of containing two adjacent code numbers that differ by only one bit. The gray code representation for the decimal numbers 0 to 15, together with the binary code is given in the Table 7.5.

7.7.3 Excess-3 Code

The excess-3 code is another BCD code used in earlier computers. The excess-3 code is not a weighted code. It is a self-complementing code and helps in performing subtraction operations in digital computers. The excess–3 code is also a reflection code.

An excess–3 code is obtained by adding 3 to each digit of a decimal number. For example, to encode the decimal number 6 into an excess–3 code, we must first add 3, in order to obtain 9. The 9 is then encoded into its equivalent 4 bit binary code 1001.

Example: Conversion of the decimal number 548 to its equivalent excess-3 code.

Decimal number	5	4	8
Add 3 *to each bit*	+3	+3	+3
Sum =	8	7	11

Hence, the equivalent excess-3code 1000 0111 1011

The representation of Excess-3 code for the decimal numbers is given in the Table 7.6.

Table 7.6: Excess-3 Code of Decimal Number

Decimal Number	Excess-3 Code
0	0011
1	0100
2	0101
3	0110
4	0111
5	1000
6	1001
7	1010
8	1011
9	1100

7.7.4 Binary to Gray Conversion

To convert a given binary number to its equivalent gray code, the following rules are applied.

1. The MSB of the gray code is same as the MSB of the binary.
2. Coding starts from left to right, add each adjacent pair of bits to get the next bit of the gray code. Omit the carry, if occurs.

Example: Conversion of the binary number 1011 to gray code.

Step1: The MSB in gray code is same as the MSB of the binary

1 0 1 1 Binary
↓
1 1 Gray

Step2: Add the left most bit to the adjacent one.

1 + 0 1 1 Binary
↓
1 1 Gray

Step3: Add the next adjacent pair.

1 0+ 1 1 Binary
↓
1 1 1 1 Gray

Step4: Add the next adjacent pair and omit the carry.

1 0 1+ 1 Binary
↓
1 1 1 0 Gray

Hence, $(1011)_2 = (1110)_G$

Suffix 'G' is used to represent the Gray code.

7.7.5 Gray to Binary Conversion

To convert a given Gray code number into equivalent binary, the following rules are applied.

1. The MSB of the binary is same as the MSB of the Gray.
2. Coding from left to right, add the binary digit generated to the adjacent gray bit to get the next bit of the binary. Omit the carry if occurs.

Example: convert the gray code 1110 to its equivalent binary.

Step1: The MSB in binary is same as the MSB of the gray

1 1 1 0 Gray
↓
1 Binary

Step2: Add the binary digit generated to the adjacent bit of the Gray code.

1 1 1 0 Gray
+
1 0 Binary

Step 3: Add the binary digit generated to the adjacent bit of Gray codes.

1 1 1 0 Gray
+
1 0 1 Binary

Step 4: Add the binary digit generated to the adjacent bit of gray code.

$$\begin{array}{cccc} 1 & 1 & 1 & 0 \\ & & + & \\ 1 & 0 & 1 & 1 \end{array}$$

Hence, $(1110)_G = (1011)_2$

7.8 ADVANTAGES AND DISADVANTAGES OF DIGITAL ELECTRONICS

7.8.1 Advantage

1. Very simple logic the lead to identify the faults very easily
2. Immune to noise
3. Flexibility of programming
4. Design and testing is very simple compared to analog electronics
5. Achieve very high speed switching

Disadvantage

1. High energy consumption than analog electronic circuits
2. Higher cost of design
3. Portability is difficult
4. Real world signals need conversion
5. Less accurate than the analog electronics

LEARNING OUTCOMES

Student will capable of

1. **Remembering of the difference between analog and digital signals.**
2. **Conversion of one number system into another number system.**
3. **Understanding of basic digital circuits.**
4. **Designing and testing of small digital application circuits.**

GLOSSARY

S. No	Terms	Explanation
1	**BCD**	Binary Coded Decimal. Four bit code used to portray each digit of a display numbers by its 4 binary equivalent
2	**Binary**	A number system having only two symbols, 0 and 1.
3	**Digital**	Relating to devices or circuits that have outputs of only two discrete levels. Example: 0 or 1, high or low, on or off, true or false etc
4	**LSB**	Least Significant Bit. Right most bit (smallest weight) of a binary expressed quantity
5	**MSB**	Most Significant Bit. Left most binary bit (largest weight) of a binary expressed quantity

QUESTIONS

PART A

I. Choose the best answer 1 Mark

1. The number of levels in a digital signals
   a) One b) Two c) Eight d) Ten

2. A sinewave is a
   a) analog signal
   b) digital signal
   c) both digital and analog signal
   d) neither digital nor analog.

3. In computer memory, 1Kilobyte is equivalent to ________ bytes?
   a) 1000 b) 1024 c) 2000 d) 3024

4. How many numbers are there in octal number?
   a) 10 b) 2 c) 8 d) 16

5. In the following numbers which one is not an OCTAL number?
   a) 56 b) 32 c) 43 d) 86

6. Decimal number 15 in binary system can be written as
   a) 1111 b) 1000 c) 1110 d) 1100

7. The equivalent Decimal value for the Binary 10101 is …………..
   a) 13 b) 19 c) 21 d) 23

8. The equivalent Octal value for the Decimal 19 is represented by
   a) 21 b) 23 c) 25 d) 22

9. The 1's complement of 0010 is ………..
   a) 0010 b) 1101 c) 0010 d) 1111

10. The 2's complement of 1110 is ………..
    a) 0010 b) 1101 c) 0010 d) 1111

II. Answer in few sentence 3 Mark

1. What is digital electronic?
2. What is binary number system?
3. What is digital signal? Give Example
4. Convert the decimal number 18 into binary number.
5. Convert the decimal number 96 into octal number.
6. Convert the decimal number 228 into hexa-decimal number.
7. Why digital system is reliable?
8. List the binary code
9. Define gray code?
10. Convert the binary number 1100 into decimal number.

III Answer the following questions with suitable examples. 5 Marks

1. Write a short note on analog and digital signals?
2. Explain Binary number system.
3. What are the advantages and disadvantages of digital electronics?
4. Convert the gray code 1110 into its equivalent binary.

IV. Answer the following questions with neat sketches. 10 Marks

1. Discuss in detail the concepts of binary codes.
2. Explain in detail about the number system.

ANSWERS

1. **(b)** 2. **(a)** 3. **(b)** 4. **(c)** 5. **(d)**
6. **(a)** 7. **(c)** 8. **(b)** 9. **(b)** 10. **(d)**